

IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Currently Amended): A ~~compiler~~ computer-readable medium storing a program for compiling for generating object code from an input source program, the object code including user-defined machine instructions defined by a user, the ~~compiler~~ program for compiling comprising:

analyzing, by a syntax analyzer, whether or not an operation described in the source program conforms to grammatical rules, outputting, by the syntax analyzer, a result of the analysis as an syntax-analysis result, and associating, by the syntax analyzer, the details of the processing operations with the user-defined machine instructions and storing the associated details of the processing operations and user-defined machine instructions in an intrinsic function definition database when detecting that the combination of the instructions is a function definition of the intrinsic function which defines the details of the processing operations associated so as to be converted into the user-defined machine instruction;

generating, by a code generator, machine instructions from the source program based on the syntax-analysis result of the syntax analyzer; and

replacing, by a code optimizer, the machine instructions by the corresponding user-defined machine instructions stored in the intrinsic function definition database in the case where the machine instructions generated by the code generator are associated with the details of the processing operations stored in the intrinsic function definition database,

wherein a statement for explicitly calling the intrinsic function is not beforehand embedded in a body of the input source program since the definitions of the intrinsic function is placed independently of the input source program, and the program for compiling is built only once since changing the user-defined machine instructions does not require for rebuilding the program for compiling.

Claim 2 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 1, the program for compiling further comprising

dividing, by a lexical analyzer, the operations described in the source program into tokens, wherein

the syntax analyzer analyzes whether or not the tokens conforms to grammatical rules, and analyzes whether or not the combination of the tokens is a function definition of the intrinsic function.

Claim 3 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 1, wherein the syntax analyzer inputs the definition of the intrinsic function and the details of the processing operations of the intrinsic function from an intrinsic function information file different from the source program.

Claim 4 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 1, wherein the definition of the intrinsic function includes information of parameter types and an identification name.

Claim 5 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 2, wherein the definition of the intrinsic function includes information of parameter types and an identification name.

Claim 6 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 3, wherein the definition of the intrinsic function includes information of parameter types and an identification name.

Claim 7 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 1, wherein in the intrinsic function definition database, plural kind of details of the processing operations can be defined for one intrinsic function.

Claim 8 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 2, wherein in the intrinsic function definition database, plural kind of details of the processing operations can be defined for one intrinsic function.

Claim 9 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 3, wherein in the intrinsic function definition database, plural kind of details of the processing operations can be defined for one intrinsic function.

Claim 10 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 4, wherein in the intrinsic function definition database, plural kind of details of the processing operations can be defined for one intrinsic function.

Claim 11 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 1, wherein the definition of the intrinsic function and the details of the processing operations of the intrinsic function can be described by C language.

Claim 12 (Currently Amended): The ~~compiler~~ computer-readable medium of claim 1, wherein the definition of the intrinsic function and the details of the processing operations of the intrinsic function can be described by hardware description language.

Claim 13 (Currently Amended): A computer implemented method of compiling for generating object code from an input source program, the object code including user-defined machine instructions defined by a user, the computer implemented method comprising:

analyzing, by a syntax analyzer, whether or not an operation described in the source program conforms to grammatical rules, outputting, by the syntax analyzer, a result of the analysis as an syntax-analysis result, and associating, by the syntax analyzer, the details of the processing operations with the user-defined machine instructions and storing the associated details of the processing operations and user-defined machine instructions in an intrinsic function definition database when detecting that the combination of the instructions is a function definition of the intrinsic function which defines the details of the processing operations associated so as to be converted into the user-defined machine instruction;

generating, by a code generator, machine instructions from the source program based on the syntax-analysis result of the syntax analyzer; and

replacing, by a code optimizer, the machine instructions by the corresponding user-defined machine instructions stored in the intrinsic function definition database in the case where the machine instructions generated by the code generator are associated with the details of the processing operations stored in the intrinsic function definition database,

wherein a statement for explicitly calling the intrinsic function is not beforehand embedded in a body of the input source program since the definitions of the intrinsic function is placed independently of the input source program, and the program for compiling is built only once since changing the user-defined machine instructions does not require for rebuilding the program for compiling.

Claim 14 (Previously Presented): The computer implemented method of claim 13, further comprising

dividing, by a lexical analyzer, the operations described in the source program into tokens, wherein

in the analyzing by the syntax analyzer, whether or not the tokens conforms to grammatical rules is analyzed, and whether or not the combination of the tokens is a function definition of the intrinsic function is analyzed.

Claim 15 (Original): The computer implemented method of claim 13, wherein the definition of the intrinsic function and the details of the processing operations of the intrinsic function are inputted, by the syntax analyzer, from an intrinsic function information file different from the source program.

Claim 16 (Original): The computer implemented method of claim 13, wherein the definition of the intrinsic function including information of parameter types and an identification name are analyzed by the syntax analyzer and stored in the intrinsic function definition database.

Claim 17 (Previously Presented): The computer implemented method of claim 13, wherein in the intrinsic function definition database, plural kind of details of the processing operations can be defined for one intrinsic function.

Claim 18 (Original): The computer implemented method of claim 13, wherein the definition of the details of the processing operations of the intrinsic function described by C language is analyzed by the syntax analyzer.

Claim 19 (Original): The computer implemented method of claim 13, wherein the definition of the intrinsic function and the details of the processing operations of the intrinsic function described by hardware description language is analyzed by the syntax analyzer.

Claim 20 (Currently Amended): A program development system for developing an application program for a processor which can execute user-defined machine instructions defined by a user, the program development system comprising:

a compile apparatus for generating object code from the application program comprising

a lexical analyzer configured to divide an operation described in a source code of the application program into tokens,

a syntax analyzer configured to analyze whether or not the tokens conform to grammatical rules, output a result of the analysis as an syntax-analysis result, and associate the details of the processing operations with the user-defined machine instructions and store the associated details of the processing operations and user-defined machine instructions in an intrinsic function definition database when detecting that the combination of the instructions is a function definition of the intrinsic function which defines the details of the processing operations associated so as to be converted into the user-defined machine instruction,

a code generator configured to generate machine instructions from the application program based on the syntax-analysis result of the syntax analyzer, and

a code optimizer configured to optimize the machine instructions by the corresponding user-defined machine instructions stored in the intrinsic function definition database in the case where the machine instructions generated by the code

generator are associated with the details of the processing operations stored in the intrinsic function definition database; and

a simulator apparatus configured to simulate the application program including the machine instruction output from the compile apparatus,

wherein a statement for explicitly calling the intrinsic function is not beforehand embedded in a body of the input source program since the definitions of the intrinsic function is placed independently of the input source program, and the program for compiling is built only once since changing the user-defined machine instructions does not require for rebuilding the program for compiling.